Markov Logic: A Unifying Framework for Statistical Relational Learning

Pedro Domingos PEDROD@CS.WASHINGTON.EDU
Matthew Richardson MATTR@CS.WASHINGTON.EDU
Department of Computer Science and Engineeering, University of Washington, Seattle, WA 98195-2350, USA

Abstract

Interest in statistical relational learning (SRL) has grown rapidly in recent years. Several key SRL tasks have been identified, and a large number of approaches have been proposed. Increasingly, a unifying framework is needed to facilitate transfer of knowledge across tasks and approaches, to compare approaches, and to help bring structure to the field. We propose *Markov logic* as such a framework. Syntactically, Markov logic is indistinguishable from first-order logic, except that each formula has a weight attached. Semantically, a set of Markov logic formulas represents a probability distribution over possible worlds, in the form of a log-linear model with one feature per grounding of a formula in the set, with the corresponding weight. We show how approaches like probabilistic relational models, knowledge-based model construction and stochastic logic programs are special cases of Markov logic. We also show how tasks like collective classification, link prediction, link-based clustering, social network modeling, and object identification can be concisely formulated in Markov logic. Finally, we briefly describe learning and inference algorithms for Markov logic, and report positive results on a link prediction task.

1. The Need for a Unifying Framework

Many (if not most) real-world application domains are characterized by the presence of both uncertainty and complex relational structure. Statistical learning focuses on the former, and relational learning on the latter. Statistical relational learning (SRL) seeks to combine the power of both. Research in SRL has expanded rapidly in recent years, both because of the need for it in applications, and because statistical and relational learning have individually matured to the point where combining them is a feasible research enterprise. A number of key SRL tasks have been identified, including collective classification, link prediction, link-based clustering, social network modeling, object identification, and others. A large and growing number of SRL approaches have been proposed, including knowledge-based model construction (Wellman et al., 1992; Ngo & Haddawy, 1997; Kersting & De Raedt, 2001), stochastic logic programs (Muggleton, 1996; Cussens, 1999), PRISM (Sato & Kameya, 1997), probabilistic relational models (Friedman et al., 1999), relational Markov models (Anderson et al., 2002), relational Markov networks (Taskar et al., 2002), relational dependency networks (Neville & Jensen, 2003), structural logistic regression (Popescul & Ungar, 2003), relational generation functions (Cumby & Roth, 2003), CLP(BN) (Costa et al., 2003), and others.

While the variety of problems and approaches in the field is valuable, it makes it difficult for researchers, students and practitioners to identify, learn and apply the essentials. In particular, for the most part, the relationships between different approaches and their relative strengths and weaknesses remain poorly understood, and innovations in one task or application do not easily transfer to others, slowing down progress. There is thus an increasingly pressing need for a unifying framework, a common language for describing and relating the different tasks and approaches. To be most useful, such a framework should satisfy the following desiderata:

1. *The framework must subsume both first-order logic and probabilistic graphical models.* Otherwise some current or future SRL approaches will

fall outside its scope.

2. *SRL problems should be representable clearly and simply in the framework.*

3. *The framework must facilitate the incorporation of domain knowledge into SRL.* Because the search space for SRL algorithms is very large even by AI standards, domain knowledge is critical to success. Conversely, the ability to incorporate rich domain knowledge is one of the most attractive features of SRL.

4. *The framework should facilitate the extension to SRL of techniques from statistical learning, inductive logic programming, probabilistic inference and logical inference.* This will speed progress in SRL by taking advantage of the large extant literature in these areas.

In the next section we propose a framework that we believe meets all of these desiderata. We then describe how several SRL approaches and tasks can be formulated in this framework. Finally, we illustrate how existing learning and inference techniques can be applied within it to yield practical algorithms.

2. Markov Logic

Markov logic is a simple yet powerful combination of Markov networks and first-order logic. Recall that a Markov network is a model for the joint distribution of a set of variables $X = (X_1, X_2, \ldots, X_n) \in \mathcal{X}$ (Pearl, 1988), often conveniently represented as a *log-linear model*:

$$P(X = x) = \frac{1}{Z} \exp\left(\sum_j w_j f_j(x)\right) \quad (1)$$

where Z is a normalization factor, and the $f_j(x)$'s are *features* of the state x (i.e., functions with $\mathcal{X}$ as the domain). Essentially every probabilistic model of interest to SRL can be represented as a Markov network or log-linear model, including Bayesian networks, decision trees, logistic regression, etc. *Markov logic* raises the expressiveness of Markov networks to encompass first-order logic. Recall that a first-order domain is defined by a set of *constants* (which we assume finite) representing objects in the domain (e.g., `Anna`, `Bob`) and a set of *predicates* representing properties of those objects and relations between them (e.g., `Smokes(x)`, `Friend(x,y)`). (For simplicity, we ignore functions in this paper; see Richardson and Domingos (2004) for a more complete treatment.) A predicate can be *grounded* by replacing its variables with constants (e.g., `Smokes(Anna)`, `Friend(Anna,Bob)`). A *world* assigns a truth value to each possible ground predicate. A first-order knowledge base (KB) is a set of formulas in first-order logic, constructed from predicates using logical connectives and quantifiers. Essentially all the relational languages used in SRL (e.g., logic programs, frame-based systems, database query languages) are special cases of first-order logic.

A formula in Markov logic is a formula in first-order logic with an associated weight. We call a set of formulas in Markov logic a *Markov logic network* or MLN. MLNs define probability distributions over possible worlds (Halpern, 1990) as follows.

Definition 2.1 *A Markov logic network L is a set of pairs (F_i, w_i), where F_i is a formula in first-order logic and w_i is a real number. Together with a finite set of constants $C = \{c_1, c_2, \ldots, c_{|C|}\}$, it defines a Markov network $M_{L,C}$ (Equation 1) as follows:*

1. *$M_{L,C}$ contains one binary node for each possible grounding of each predicate appearing in L. The value of the node is 1 if the ground predicate is true, and 0 otherwise.*

2. *$M_{L,C}$ contains one feature for each possible grounding of each formula F_i in L. The value of this feature is 1 if the ground formula is true, and 0 otherwise. The weight of the feature is the w_i associated with F_i in L.*

A first-order KB can be seen as a set of hard constraints on the set of possible worlds: if a world violates even one formula, it has zero probability. The basic idea in Markov logic is to soften these constraints: when a world violates one formula in the KB it is less probable, but not impossible. The fewer formulas a world violates, the more probable it is. A formula's associated weight reflects how strong a constraint it is: the higher the weight, the greater the difference in log probability between a world that satisfies the formula and one that does not, other things being equal. As weights increase, an MLN increasingly resembles a purely logical KB. In the limit of all infinite weights, the MLN represents a uniform distribution over the worlds that satisfy the KB.

An MLN without variables (i.e., containing only ground formulas) is an ordinary Markov network. Any log-linear model over Boolean variables can be represented as an MLN, since each state of a Boolean clique is defined by a conjunction of literals. (This extends trivially to discrete variables, and to binary encoding of numeric variables.)

An MLN can be viewed as a *template* for constructing Markov networks. In different worlds (different sets of constants) it will produce different networks, and these may be of widely varying size, but all will have certain regularities in structure and parameters, given by the MLN (e.g., all groundings of the same formula will have the same weight).

3. SRL Approaches

Since Markov logic subsumes first-order logic and probabilistic graphical models, it subsumes all representations used in SRL that are formed from special cases of them. However, it is enlightening to see how these representations map into Markov logic, and here we informally do this for a few of the most popular ones.

3.1. Knowledge-Based Model Construction

Knowledge-based model construction (KBMC), the oldest SRL approach (Wellman et al., 1992; Ngo & Haddawy, 1997; Kersting & De Raedt, 2001), is a combination of logic programming and Bayesian networks. KBMC, like all other SRL approaches based on logic programming, is a restriction of Markov logic to KBs containing only Horn clauses. As in Markov logic, nodes in KBMC represent ground predicates. The parents of a node are the predicates appearing in the bodies of Horn clauses having the node as a consequent. The conditional probability of a node given the truth values of its parent rule bodies is specified by a combination function (e.g., noisy OR, logistic regression, arbitrary CPT). A KBMC model is translated into Markov logic by writing down a set of formulas for each first-order predicate Pk(...) in the domain. Each formula is a conjunction containing Pk(...) and one literal per parent of Pk(...) (i.e., per first-order predicate appearing in a Horn clause having Pk(...) as the consequent). A subset of these literals are negated; there is one formula for each possible combination of positive and negative literals. The weight of the formula is $w = \log[p/(1-p)]$, where p is the conditional probability of the child predicate when the corresponding conjunction of parent literals is true, according to the combination function used. If the combination function is logistic regression, it can be represented using only a linear number of formulas, taking advantage of the fact that it is a (conditional) Markov network with a binary clique between each predictor and the response. Noisy OR can similarly be represented with a linear number of parents.

3.2. Stochastic Logic Programs

Stochastic logic programs (SLPs) (Muggleton, 1996; Cussens, 1999) are a combination of logic programming and log-linear models. Puech and Muggleton (2003) showed that SLPs are a special case of KBMC, and thus they can be represented in Markov logic in the same way.

3.3. Probabilistic Relational Models

Probabilistic relational models (PRMs) (Friedman et al., 1999) are a combination of frame-based systems and Bayesian networks. PRMs can be represented in Markov logic by defining a predicate $\mathtt{S}(\mathtt{x},\mathtt{v})$ for each (propositional or relational) attribute of each class, where $\mathtt{S}(\mathtt{x},\mathtt{v})$ means "The value of attribute S in object x is v." A PRM is then translated into Markov logic by writing down a formula for each line of each (class-level) conditional probability table (CPT) and value of the child attribute. The formula is a conjunction of literals stating the parent values and a literal stating the child value, and its weight is the logarithm of $P(x|Parents(x))$, the corresponding entry in the CPT. In addition, the MLN contains formulas with infinite weight stating that each attribute must take exactly one value. Notice that this approach handles all types of uncertainty in PRMs (attribute, reference and existence uncertainty).

3.4. Relational Markov Networks

Relational Markov networks (RMNs) (Taskar et al., 2002) are a combination of Markov networks and conjunctive queries, a subset of the SQL database query language. An RMN is simply an MLN with a formula (in particular, a conjunction of literals) for each possible state of each clique template in the RMN, with the corresponding weight.

3.5. Structural Logistic Regression

In structural logistic regression (SLR) (Popescul & Ungar, 2003), the predictors are the output of SQL queries over the input data. Just as a logistic regression model is a discriminatively-trained Markov network, an SLR model is a discriminatively-trained MLN.[1]

3.6. Relational Dependency Networks

In a relational dependency network (RDN), each node's probability conditioned on its Markov blanket is

[1]Use of SQL aggregates requires that their definitions be imported into the MLN.

given by a decision tree (Neville & Jensen, 2003). Every RDN has a corresponding MLN in the same way that every dependency network has a corresponding Markov network, given by the stationary distribution of a Gibbs sampler operating on it (Heckerman et al., 2000).

4. SRL Tasks

In this section, we show how key SRL tasks can be concisely formulated in Markov logic, making it possible to bring the full power of logical and statistical learning and inference approaches to bear on them.

4.1. Collective Classification

The goal of ordinary classification is to predict the class of an object given its attributes. In collective classification, we also take into account the classes of related objects. Attributes can be represented in Markov logic as predicates of the form $\mathtt{A}(\mathtt{x}, \mathtt{v})$, where $\mathtt{A}$ is an attribute, $\mathtt{x}$ is an object, and $\mathtt{v}$ is the value of $\mathtt{A}$ in $\mathtt{x}$. The class is a designated attribute $\mathtt{C}$, representable by $\mathtt{C}(\mathtt{x}, \mathtt{v})$, where $\mathtt{v}$ is $\mathtt{x}$'s class. Classification is now simply the problem of inferring the truth value of $\mathtt{C}(\mathtt{x}, \mathtt{v})$ for all $\mathtt{x}$ and $\mathtt{v}$ of interest given all known $\mathtt{A}(\mathtt{x}, \mathtt{v})$. Ordinary classification is the special case where $\mathtt{C}(\mathtt{xi}, \mathtt{v})$ and $\mathtt{C}(\mathtt{xj}, \mathtt{v})$ are independent for all $\mathtt{xi}$ and $\mathtt{xj}$ given the known $\mathtt{A}(\mathtt{x}, \mathtt{v})$. In collective classification, the Markov blanket of $\mathtt{C}(\mathtt{xi}, \mathtt{v})$ includes other $\mathtt{C}(\mathtt{xj}, \mathtt{v})$, even after conditioning on the known $\mathtt{A}(\mathtt{x}, \mathtt{v})$. Relations between objects are represented by predicates of the form $\mathtt{R}(\mathtt{xi}, \mathtt{xj})$. A number of interesting generalizations are readily apparent, for example $\mathtt{C}(\mathtt{xi}, \mathtt{v})$ and $\mathtt{C}(\mathtt{xj}, \mathtt{v})$ may be indirectly dependent via unknown predicates, possibly including the $\mathtt{R}(\mathtt{xi}, \mathtt{xj})$ predicates themselves. Background knowledge can be incorporated by stating it in first-order logic, learning weights for the resulting formulas, and possibly refining them (see Richardson and Domingos (2004) for an example).

4.2. Link Prediction

The goal of link prediction is to determine whether a relation exists between two objects of interest (e.g., whether `Anna` is `Bob`'s Ph.D. advisor) from the properties of those objects and possibly other known relations. The formulation of this problem in Markov logic is identical to that of collective classification, with the only difference that the goal is now to infer the value of $\mathtt{R}(\mathtt{xi}, \mathtt{xj})$ for all object pairs of interest, instead of $\mathtt{C}(\mathtt{x}, \mathtt{v})$.

4.3. Link-Based Clustering

The goal of clustering is to group together objects with similar attributes. In model-based clustering, we assume a generative model $P(X) = \sum_C P(C)P(X|C)$, where X is an object, C ranges over clusters, and $P(C|X)$ is X's degree of membership in cluster C. In link-based clustering, objects are clustered according to their links (e.g., objects that are more closely related are more likely to belong to the same cluster), and possibly according to their attributes as well. This problem can be formulated in Markov logic by postulating an unobserved predicate $\mathtt{C}(\mathtt{x}, \mathtt{v})$ with the meaning "`x` belongs to cluster `v`," and having formulas in the MLN involving this predicate and the observed ones (e.g., $\mathtt{R}(\mathtt{xi}, \mathtt{xj})$ for links and $\mathtt{A}(\mathtt{x}, \mathtt{v})$ for attributes). Link-based clustering can now be performed by learning the parameters of the MLN, and cluster memberships are given by the probabilities of the $\mathtt{C}(\mathtt{x}, \mathtt{v})$ predicates conditioned on the observed ones.

4.4. Social Network Modeling

Social networks are graphs where nodes represent social actors (e.g., people) and arcs represent relations between them (e.g., friendship). Social network analysis (Wasserman & Faust, 1994) is concerned with building models relating actors' properties and their links. For example, the probability of two actors forming a link may depend on the similarity of their attributes, and conversely two linked actors may be more likely to have certain properties. These models are typically Markov networks, and can be concisely represented by formulas like $\forall \mathtt{x} \forall \mathtt{y} \forall \mathtt{v}\ \mathtt{R}(\mathtt{x}, \mathtt{y}) \Rightarrow (\mathtt{A}(\mathtt{x}, \mathtt{v}) \Leftrightarrow \mathtt{A}(\mathtt{y}, \mathtt{v}))$, where $\mathtt{x}$ and $\mathtt{y}$ are actors, $\mathtt{R}(\mathtt{x}, \mathtt{y})$ is a relation between them, $\mathtt{A}(\mathtt{x}, \mathtt{v})$ represents an attribute of $\mathtt{x}$, and the weight of the formula captures the strength of the correlation between the relation and the attribute similarity. For example, a model stating that friends tend to have similar smoking habits can be represented by the formula $\forall \mathtt{x} \forall \mathtt{y}\ \mathtt{Friends}(\mathtt{x}, \mathtt{y}) \Rightarrow (\mathtt{Smokes}(\mathtt{x}) \Leftrightarrow \mathtt{Smokes}(\mathtt{y}))$. Notice that this formula is false as a universally quantified statement in first-order logic, but is true in some domains as a probabilistic statement in Markov logic (Lloyd-Richardson et al., 2002). As well as encompassing existing social network models, Markov logic allows richer ones to be easily stated (e.g., by writing formulas involving multiple types of relations and multiple attributes, as well as more complex dependencies between them). These models can then be learned and applied using techniques like those in Richardson and Domingos (2004) (see next section).

4.5. Object Identification

Object identification (also known as record linkage, de-duplication, and others) is the problem of determining which records in a database refer to the same real-world entity (e.g., which entries in a bibliographic database represent the same publication). This problem is of crucial importance to many companies, government agencies, and large-scale scientific projects. One way to represent it in Markov logic is by defining a predicate $\texttt{Same}(\texttt{x},\texttt{y})$ with the meaning "x represents the same real-world entity as y." This predicate is applied both to records and their fields (e.g., Same("ICML", "Intl. Conf. on Mach. Learn.")). The dependencies between record matches and field matches can then be represented by formulas like $\forall \texttt{x} \forall \texttt{y}\ \texttt{Same}(\texttt{x},\texttt{y}) \Leftrightarrow \texttt{Same}(\texttt{fi}(\texttt{x}),\texttt{fi}(\texttt{y}))$, where x and y are records and $\texttt{fi}(\texttt{x})$ is a function returning the value of the ith field of record x. We have successfully applied this approach to de-duplicating the Cora database of computer science papers (Parag & Domingos, 2004). Because it allows information to propagate from one match decision (i.e., one grounding of $\texttt{Same}(\texttt{x},\texttt{y})$) to another via fields that appear in both pairs of records, it effectively performs collective object identification, and in our experiments outperformed the traditional method of making each match decision independently of all others. For example, matching two references may allow us to determine that "ICML" and "MLC" represent the same conference, which in turn may help us to match another pair of references where one contains "ICML" and the other "MLC." Markov logic also allows additional information to be incorporated into a de-duplication system easily, modularly and uniformly. For example, transitive closure is incorporated by adding the formula $\forall \texttt{x} \forall \texttt{y} \forall \texttt{z}\ \texttt{Same}(\texttt{x},\texttt{y}) \wedge \texttt{Same}(\texttt{y},\texttt{z}) \Rightarrow \texttt{Same}(\texttt{x},\texttt{z})$, with a weight that can be learned from data.

5. Implementation

In principle, any inductive logic programming (ILP) approach can be used to learn the structure of an MLN, and any approach for learning Markov network parameters (e.g., conjugate gradient or iterative scaling) can be used to learn the weights. Likewise, any method for inference in Markov networks (e.g., Markov chain Monte Carlo, belief propagation) can be used to perform inference in grounded MLNs, and logical inference methods can be used to construct the subsets of these networks relevant to a particular query. Logical inference can also be used to find modes of the distribution, which, if the KB is satisfiable and all weights are positive, are the satisfying assignments of truth values to ground predicates. When no satisfying assignments exist, modes can still be found using methods like MaxWalkSat, a variation of the WalkSat satisfiability search algorithm for finding truth assignments that maximize the sum of weights of satisfied clauses (Selman et al., 1996).

In Richardson and Domingos (2004), we describe one possible implementation of Markov logic, using MaxWalkSat and Gibbs sampling for inference, the CLAUDIEN ILP system (De Raedt & Dehaspe, 1997) for structure learning, and a pseudo-likelihood method for parameter learning (Besag, 1975). We have tested this approach on a link prediction task (predicting which students are advised by which faculty from a multi-relational database describing our department), and found that it outperforms a purely relational learner (CLAUDIEN), a purely statistical learner (Bayesian networks, restricted or not to a naive Bayes structure), and a pure knowledge-based approach (manually constructed first-order KB).

6. Conclusion

The rapid growth in the variety of SRL approaches and tasks has led to the need for a unifying framework. In this paper we propose *Markov logic* as a candidate for such a framework. Markov logic subsumes first-order logic and Markov networks, and allows a wide variety of SRL tasks and approaches to be formulated in a common language. Initial experiments with an implementation of Markov logic have yielded good results.

Acknowledgements

We are grateful to Julian Besag, Nilesh Dalvi, Alon Halevy, Henry Kautz, Tian Sang, Bart Selman, Dan Suciu, and Wei Wei for helpful discussions. This research was partly supported by an IBM Ph.D. Fellowship to the first author, and by ONR grant N00014-02-1-0408.

References

Anderson, C., Domingos, P., & Weld, D. (2002). Relational Markov models and their application to adaptive Web navigation. *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 143–152). Edmonton, Canada: ACM Press.

Besag, J. (1975). Statistical analysis of non-lattice data. *The Statistician*, *24*, 179–195.

Costa, V. S., Page, D., Qazi, M., , & Cussens, J.

(2003). CLP(BN): Constraint logic programming for probabilistic knowledge. *Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence* (pp. 517–524). Acapulco, Mexico: Morgan Kaufmann.

Cumby, C., & Roth, D. (2003). Feature extraction languages for propositionalized relational learning. *Proceedings of the IJCAI-2003 Workshop on Learning Statistical Models from Relational Data* (pp. 24–31). Acapulco, Mexico: IJCAII.

Cussens, J. (1999). Loglinear models for first-order probabilistic reasoning. *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence* (pp. 126–133). Stockholm, Sweden: Morgan Kaufmann.

De Raedt, L., & Dehaspe, L. (1997). Clausal discovery. *Machine Learning*, *26*, 99–146.

Friedman, N., Getoor, L., Koller, D., & Pfeffer, A. (1999). Learning probabilistic relational models. *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence* (pp. 1300–1307). Stockholm, Sweden: Morgan Kaufmann.

Halpern, J. (1990). An analysis of first-order logics of probability. *Artificial Intelligence*, *46*, 311–350.

Heckerman, D., Chickering, D. M., Meek, C., Rounthwaite, R., & Kadie, C. (2000). Dependency networks for inference, collaborative filtering, and data visualization. *Journal of Machine Learning Research*, *1*, 49–75.

Kersting, K., & De Raedt, L. (2001). Towards combining inductive logic programming with Bayesian networks. *Proceedings of the Eleventh International Conference on Inductive Logic Programming* (pp. 118–131). Strasbourg, France: Springer.

Lloyd-Richardson, E., Kazura, A., Stanton, C., Niaura, R., & Papandonatos, G. (2002). Differentiating stages of smoking intensity among adolescents: Stage-specific psychological and social influences. *Journal of Consulting and Clinical Psychology*, *70*.

Muggleton, S. (1996). Stochastic logic programs. In L. de Raedt (Ed.), *Advances in inductive logic programming*, 254–264. Amsterdam, Netherlands: IOS Press.

Neville, J., & Jensen, D. (2003). Collective classification with relational dependency networks. *Proceedings of the Second International Workshop on Multi-Relational Data Mining* (pp. 77–91). Washington, DC: ACM Press.

Ngo, L., & Haddawy, P. (1997). Answering queries from context-sensitive probabilistic knowledge bases. *Theoretical Computer Science*, *171*, 147–177.

Parag, & Domingos, P. (2004). *Collective record linkage* (Technical Report). Department of Computer Science and Engineering, University of Washington, Seattle, WA. http://www.cs.washington.edu/-homes/pedrod/crl.pdf.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: Networks of plausible inference*. San Francisco, CA: Morgan Kaufmann.

Popescul, A., & Ungar, L. H. (2003). Structural logistic regression for link analysis. *Proceedings of the Second International Workshop on Multi-Relational Data Mining* (pp. 92–106). Washington, DC: ACM Press.

Puech, A., & Muggleton, S. (2003). A comparison of stochastic logic programs and Bayesian logic programs. *Proceedings of the IJCAI-2003 Workshop on Learning Statistical Models from Relational Data* (pp. 121–129). Acapulco, Mexico: IJCAII.

Richardson, M., & Domingos, P. (2004). *Markov logic networks* (Technical Report). Department of Computer Science and Engineering, University of Washington, Seattle, WA. http://www.cs.washington.-edu/homes/pedrod/mln.pdf.

Sato, T., & Kameya, Y. (1997). PRISM: A symbolic-statistical modeling language. *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence* (pp. 1330–1335). Nagoya, Japan: Morgan Kaufmann.

Selman, B., Kautz, H., & Cohen, B. (1996). Local search strategies for satisfiability testing. In D. S. Johnson and M. A. Trick (Eds.), *Cliques, coloring, and satisfiability: Second DIMACS implementation challenge*, 521–532. Washington, DC: American Mathematical Society.

Taskar, B., Abbeel, P., & Koller, D. (2002). Discriminative probabilistic models for relational data. *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence* (pp. 485–492). Edmonton, Canada: Morgan Kaufmann.

Wasserman, S., & Faust, K. (1994). *Social network analysis: Methods and applications*. Cambridge, UK: Cambridge University Press.

Wellman, M., Breese, J. S., & Goldman, R. P. (1992). From knowledge bases to decision models. *Knowledge Engineering Review*, *7*.